

Why Linux

Является продуктом мирового сообщества программистов и заинтересованных пользователей, участвующих в сопровождении множества разновидностей и версий этой операционной системы.

Исходный код открыт, и любой желающий, обладающий достаточной квалификацией, может самостоятельно собрать ядро, проведя кастомизацию ОС для каких-либо собственных целей, например, при создании собственной программно-аппаратной системы. Использование только тех компонентов ядра, что необходимы.

Никакие trackers и back doors не могут быть оставлены.

О сертификации ПАК.

Трудозатраты сертификации соизмеримы с трудоемкостью разработки самого изделия.

Стабильность работы системы

Долговременная работа без перезагрузок.

Обновление системы (патчи и минорные обновления ОС) прямо на ходу без перезапусков.

LTS (Long-term support) версии ОС выпускаются обычно раз в год-два.

Linux применяется и адаптируется для все новых целей. Уже давно, не только на серверах и рабочих станциях и мобильных устройствах, но и на роутерах, в самых различных ПАК, управляющими системами связи, технологическими линиями, контролирующими движение, в IoT и т.д.

Linux based servers are the backbone of the Internet.

Одной из причин того, что ядро Linux столь удачно и стабильно, считается то, что его создатель Linus Torvalds заложил основы архитектуры единолично (по существу являлся диктатором).

Linux может использоваться где угодно, потому что это free-as-in-freedom, extremely reliable and predictable, costs nothing.

Ядро Linux настолько удачно, что даже у компании Майкрософт была замечена тенденция движения в сторону изыскания возможностей замещения собственного текущего ядра, ядром подобным Linux. Естественно с оставлением всего окружения виндоусподобным, то есть таким как есть. Тем более Майкрософт не приемлет Линуксовский оконный менеджер X11.

Такое замещение ядра, возможно, вдохнет в ОС Windows новую жизнь и позволит ей еще долго оставаться на ведущих позициях, в качестве офисной ОС, а также ОС для домашних применений.

Известно название такой версии Windows - это Redmond.

Не факт, что это случится, однако компания Apple уже проделала подобное, но не совсем с ядром Linux, а с его родственником версии BSD Unix. И чувствует себя превосходно.

Компания Майкрософт сближается с Linux и по ряду программных продуктов, например, широко используемый для дистанц обучения и вебинаров, продукт Microsoft Teams

Slide about SW substitution

SW: Win <==> Lin

Даже обработка видео Kazam , OBS Studio

Графика: Photoshop - Gimp

Среды разработки: VisualStudio - Eclipse, QT4

MS Office <==> OpenOffice (LibreOffice) (пока табуляция и форматирование не на 100% совместимо, съезжает иногда)

Some popular SW like Punto Switcher, Lingvo Dictionary still unsatisfactorily

Linux distros

Tweaker-ы помогут сделать многое в графич интерфейсе.

Но часто различные дистрибутивы, как раз освобождают от этих настроек, имея вполне подходящий функционал и внешний вид GUI.

Ubuntu – наибольшее количество загрузок дистрибутива.

Lubuntu – для железа послабее.

Mint – прост, ничего лишнего.

Fedora – экономичный расход акк-ра ноутбука.

Debian, Gentu (быстр) - под сервера

Используем Ubuntu (весной вышел Long Term Support релиз 20.04)

Distros domestically produced

выполнения нового вида профессиональной деятельности с использованием компьютеров, работающих под открытой ОС Linux, а также российских персональных компьютеров для

государственных предприятий, работающих под российскими ОС, собранными на ядре Linux, такими, как

Astra Linux, MCBC, GosLinux, «Альт 8 СП», «Ось», Sailfish, ROSA Linux, AlterOS, Calculate Linux, ICLinux и многими другими.

Применение отечественных ОС позволяет отказаться от использования иностранного ПО, включая ОС иностранного производства типа Windows или Mac OS, которые, как известно, закрывая свой код, не дают никаких гарантий безопасности. Это для госучреждений и military.

Like импортозамещение.

Достаточно China&Malaysia отказ от Win в госучреждениях => MS kaput.

Resources

Наиболее полная информация - в Интернете.
Тематика Linux и сети – иначе и быть не может.

На ютубе множество учебных роликов, видео лекций.
В случае нашего перехода на дистанционный режим обучения,
подобные ролики подключатся к ресурсам вашей системы
дистанционного образования.

Kernel structure

Основные подсистемы ядра:

1. Файловая подсистема
2. Подсистема управления процессами и памятью
3. Подсистема ввода/вывода
4. Сетевая подсистема

File System file types

В UNIX существуют 6 типов файлов,
различающихся по функциональному назначению и действиям ОС при
выполнении операций над файлами:

- regular file - обычный файл
- directory - каталог
- special device file - специальный файл устройства
- named pipe - FIFO или именованный канал
- link - связь
- socket – сокет

File System inode

Каждый файл имеет связанные с ним метаданные,
хранящиеся в индексных дескрипторах – **inode** (*index node*),
содержащие все характеристики файла
и позволяющие операционной системе выполнять с ним операции,
заказанные прикладной задачей:

открыть файл,
 прочитать или
 записать данные,
 создать или
 удалить файл.

В частности, метаданные содержат и указатели на дисковые блоки
хранения данных файла.

Каталог – это файл, содержащий имена находящихся в нем файлов,

а также указатели на дополнительную информацию - *метаданные*, (информация для операционной системы при работе с файлом).
Каталоги определяют положение файла в дереве файловой системы,

поскольку сам файл никак не содержит информации о своем местонахождении.

По существу каталог представляет собой таблицу, каждая запись которой соответствует некоторому файлу. Первое поле каждой записи содержит указатель на метаданные (номер *индексного дескриптора inode*), а второе - определяет имя файла.

Иноды – это данные о данных или метаданные, как любят выражаться айтишники.

Линуксная инода содержит достаточно много метаданных:

- размер файла
 - идентификатор (ID) устройства, содержащего файл
 - ID пользователя-владельца файла
 - ID группы пользователей, к которой принадлежит владелец
 - права доступа, присвоенные владельцу, группе и т.н. «другим»
 - права чтения, модификации и исполнения для владельца, группы и других
 - таймштампы моментов последнего доступа и изменения файла
 - счётчик количества жёстких ссылок на файл (если он не нулевой, то файл невозможно удалить)
 - указатели на блоки (кластеры) диска, в которых размещён файл
 - имя папки или блочного устройства, где расположен файл
 - размер блока файловой системы (ФС) раздела, в котором находится файл
 - количество блоков, занимаемых файлом
 - инода не содержит ни таймштампа создания файла, ни его имени.
- Это атрибуты таблицы директориев (**directory table**).

Когда пользователь пытается получить к файлу доступ, происходит следующее:

- по имени файла в таблице директориев определяется соответствующий ему номер иноды,
- по номеру иноды происходит обращение к **Inode table** и считываются метаданные,
- начинается работа с файлом, если это возможно.

File System links

Связь. Каталог содержит имена файлов и указатели на их метаданные. В то же время, сами метаданные не содержат ни имени файла, ни указателя на это имя. Такая архитектура позволяет одному файлу иметь несколько имен в файловой системе.

Имена жестко связаны с метаданными, в то время как сам файл существует независимо от того, как его называют в файловой системе.

Можно создать еще одно (и более) имен одного и того же файла. В списках файлов каталогов, файлы будут отличаться только именем. Все остальные *атрибуты файла* будут абсолютно одинаковыми. С точки зрения пользователя - это два разных файла, но изменения, внесенные в любой из этих файлов, затронут и другой, поскольку оба они ссылаются на одни и те же данные. Удаление **одного из** файлов **не приведет** к удалению самого файла, т. е. его метаданных и данных.

Количество ссылок (разных имен одного и того же файла) указывается в соответствующем поле списка файлов каталога.

File System links symbolic

Помимо организации *жесткой связи* возможно Установление *символической связи (гибкой ссылки)* с помощью специального файла, который будет только косвенно адресовать целевой файл. Данные специального файла (являющегося символической ссылкой) содержат только полное имя целевого файла, то есть ссылаются на него.

File System Sockets

Сокеты : Этот тип файлов предназначен для организации взаимодействия между процессами исполняющимися на разных компьютерах. Интерфейс сокетов используется для доступа к сети на IP протоколе.

С каждым сокет связываются три атрибута: **домен, тип и протокол**. Эти атрибуты задаются при создании сокета и остаются неизменными на протяжении всего времени его существования.

Домен определяет пространство адресов, в котором располагается сокет, и множество протоколов, которые используются для передачи данных. Чаще других используются домены Unix и Internet.

При задании AF_UNIX для передачи данных используется файловая система ввода/вывода Unix. В этом случае сокеты используются для межпроцессного взаимодействия на одном компьютере и не годятся для работы по сети.

Константа AF_INET соответствует Internet-домени.

Сокеты, размещённые в этом домене, могут использоваться для работы в любой IP-сети.

Существуют и другие домены (AF_IPX для протоколов Novell, AF_INET6 для новой модификации протокола IP - IPv6 и т. д.).

- **SOCK_STREAM.** Передача потока данных с предварительной установкой соединения. Обеспечивается надёжный канал передачи данных, при котором фрагменты отправленного блока не теряются, не переупорядочиваются и не дублируются. Этот тип сокетов является самым распространённым
- **SOCK_DGRAM.** Передача данных в виде отдельных сообщений (датаграмм). Предварительная установка соединения не требуется. Обмен данными происходит быстрее, но является ненадёжным: сообщения могут теряться в пути, дублироваться и переупорядочиваться. Допускается передача сообщения нескольким получателям (multicasting) и широковещательная передача (broadcasting).
- **SOCK_RAW.** Этот тип присваивается низкоуровневым (т. н. "сырым") сокетам. Их отличие от обычных сокетов состоит в том, что с их помощью программа может взять на себя формирование некоторых заголовков, добавляемых к сообщению.

Не все домены допускают задание произвольного типа сокета. Например, совместно с доменом Unix используется только тип **SOCK_STREAM**. С другой стороны, для Internet-домена можно задавать любой из перечисленных типов. В этом случае для реализации **SOCK_STREAM** используется протокол TCP, для реализации **SOCK_DGRAM** - протокол UDP, а тип **SOCK_RAW** используется для низкоуровневой работы с протоколами IP, ICMP и т. д.

File System structure

Файлы в UNIX организованы в виде древовидной структуры. Корнем этого дерева является корневой каталог (root directory), имеющий имя "/" . Полное имя любого файла начинается с "/" и не содержит идентификатор устройства (дискового накопителя), на котором он фактически хранится (Win).

Использование общепринятых имен основных файлов и структуры каталогов существенно облегчает работу в UNIX, ее **администрирование** и переносимость. Нарушение этой структуры может привести к неработоспособности системы или отдельных ее компонентов.

Корневой каталог / является основой любой файловой системы UNIX. Все файлы и каталоги располагаются в рамках структуры, порожденной корневым каталогом, независимо от их физического местонахождения.

В каталоге **/bin** находятся наиболее часто употребляемые команды и утилиты системы.

Каталог **/dev** содержит специальные файлы устройств, являющиеся интерфейсом доступа к периферийным устройствам.

/etc . В этом каталоге находятся системные конфигурационные файлы и многие утилиты администрирования, например, скрипты инициализации системы.

В каталоге **/lib** находятся библиотечные файлы языка C и других языков программирования. Стандартные названия библиотечных файлов имеют вид типа `libx.a` или `libx.so`. Часть библиотечных файлов также находится в каталоге **/usr/lib**

/lost+found каталог "потерянных" файлов. При аппаратных сбоях могут возникать ошибки целостности файловой системы, что может приводить к появлению «безымянных» файлов. Структура и содержимое такого файла являются правильными, однако для него отсутствует имя в каком-либо из каталогов. Программы проверки и восстановления файловой системы помещают такие файлы в каталог **/lost+found** под системными именами (числовыми).

/mnt стандартный каталог для временного связывания (монтирования) **физических** файловых систем к **корневой** для получения единого дерева логической файловой системы.

В каталоге **/usr** находятся подкаталоги различных сервисных подсистем, исполняемые файлы утилит UNIX, подкаталоги электронной почты, а также и дополнит. пользовательские программы, библиотеки т. д.

Каталог **/var** используется для хранения временных файлов различных сервисных подсистем системы печати и т. д.

Каталог **/tmp** предназначен для хранения временных файлов, необходимых для работы различных подсистем UNIX, а также файлов пользователей. Этот каталог открыт на запись для всех пользователей системы, в том числе для работающих под гостевым аккаунтом **Guest account**.

/home каталог для размещения домашних каталогов пользователей. Например, имя домашнего каталога пользователя alex будет называться `/home/alex` .

Для пользователя, работающего под гостевым аккаунтом, не создается соответствующего ему каталога в **/home** и после окончания сеанса работы от гостя не остается никаких файлов.

Владельцы файлов

Файлы в UNIX имеют два типа владельцев: пользователя (user owner) и группу (group owner).

Группой называется определенный список пользователей системы. Пользователь системы может быть членом нескольких групп : одна из которых является *первичной (primary)*, остальные - *дополнительными (supplementary)*.

При этом что владелец-пользователь может не являться членом группы, владеющей файлом. Это дает большую гибкость в организации доступа к файлам.

Совместное пользование файлами можно организовать практически для любого состава пользователей, создав соответствующую группу и установив для нее права на соответствующие файлы.

То есть для того чтобы некий пользователь получил доступ к этим файлам, достаточно включить его в группу-владельца.

На этапе создания файла его владельцем-пользователем становится тот пользователь, который его создает. Определить, владельцев файлов можно, исполнив команду **ls -l**. Вид содержимого текущего каталога:

```
1      2 3  4      5      6  7  8
- rw- r- - r-- 1 andy group 235520 Dec 22 19:15 file1.txt
- rw- rwx r-- 1 andy student 3450 Nov 12 19:15 file2.txt
```

Третья и четвертая колонки содержат имена владельца-пользователя и владельца-группы.

Владение файлом определяет тот набор операций, который пользователь может совершить с файлом.

Изменение владельца файла или прав доступа к файлу может осуществлять только владелец файла или суперпользователь (su).

Права доступа к файлу контролируют такие операции, как чтение, запись и запуск на выполнение (для исполняемых файлов).

Права доступа к файлам

Linux существуют три базовых *класса доступа* к файлу, в каждом из которых установлены соответствующие права доступа:

User access (**u**) для владельца-пользователя файла

Group access (**g**) для членов группы, являющейся владельцем файла

Other access (**o**) для остальных пользователей .

UNIX поддерживает *три типа прав доступа* для каждого класса:
на чтение (read), обозначается символом **r** ,
на запись (write), обозначается символом **w** ,
на выполнение (execute), обозначается символом **x** .
Отсутствие прав доступа отображается символом - .

Например, права доступа к файлу a.out :

- rwx r-x r-- 1 andy student 4889 Nov 10 19:15 a.out

Тип файла	Права владельца- пользователя	Права владельца- группы	Права остальных пользователей
Обычный	Чтение, запись, выполнение	Чтение и выполнение	Только чтение

Для изменения прав доступа используется команда **chmod** .

\$ chmod u+w, og+r- w file1.txt file2.txt

Добавить право записи для владельца,
и право на чтение для группы и остальных.
Отключить право на запись для всех пользователей,
исключая владельца.